Amendments to the Drawings:

Please delete the drawing sheet containing Figure 6, which was submitted in the Amendment dated June 21, 2010.

REMARKS

Claims 1-7, 11-13, 15-18, 27-29 and 31-33 are pending in the application. Claims 8-10, 14, 19-26, 30 and 34 were previously canceled. No new matter is introduced.

The Examiner objected to Figure 6 and the description thereof added to the specification in the amendment of June 21, 2010 as allegedly introducing new matter. While Applicants disagree with the Examiner's contention that Figure 6 and the description thereof constitute new matter, Figure 6 and the description thereof (which Applicants' have consistently contended are not necessary, but which were added in response to an objection by the Examiner) are deleted in the present amendment to expedite prosecution. Applicants reserve the right to reintroduce these amendments, for example in a further amendment, and to petition for review of the Examiner's objection should the Examiner renew the objection. It is noted that the Examiner did not explain why Figure 6 and the description thereof allegedly introduced new matter, and did not address Applicants' contention that Figure 6 and the description thereof were not new matter in view of original claim 1. For Example, the Examiner appears to contend that copying the original language of claims 1 and 2 into the specification constituted introducing new matter. Applicants do not believe this is a reasonable interpretation of the prohibition against introducing new matter.

This Amendment is supported by the previously filed Declaration of Dr. Kaushik Saha and Mr. Srijib Narayan Maiti (hereinafter "Inventor Decl."). Dr. Saha and Mr. Maiti are experts in the field of signal processing. Inventor Decl., ¶¶ 1-2, Appendixes 1 and 2. Dr. Saha and Mr. Maiti have reviewed the present application, the Office Action mailed on February 1, 2010, and the references cited therein. Inventor Decl., ¶ 3. It is noted that the Examiner does not appear to have considered the declaration at all in the Final Rejection.

The Drawings Are Sufficient

The Examiner maintained, almost verbatim and without addressing Applicants' arguments or the details of proffered Figure 6 of the previous amendment, the Examiner's previous objection to the drawings under 35 CFR Section 1.83(a) for failing to show all the limitations of the claims. Specifically, the Examiner contends the figures do not show the "structure of computing/performing N-point FFT/IFFT of the signal using first and second stages

wherein the second stage employs single, un-nested computational loop," of the independent claims. The Examiner's objections are respectfully traversed.

As noted in the previous amendment, the independent claims do not use the exact language about which the Examiner complains. The Examiner did not address this point in the Final Office Action, or explain why the proffered Figure 6 which was added in the last amendment failed to address the Examiner's concerns or provide any indication of what corrective action was considered necessary by the Examiner to the other figures. Before turning to the language of each independent claim, it is noted that, for computer-related claims, a disclosure of a processor capable of performing a function and the function is generally sufficient. *See, e.g.*, *Fonar Corp. v. General Electric Corp.*, 107 F.3d 1543, 1549 (Fed. Cir. 1997) ("As a general rule, where software constitutes part of a best mode of carrying out an invention, description of such a best mode is satisfied by a disclosure of the function of the software. This is because, normally, writing code for such software is within the skill of the art, not requiring undue experimentation, once its functions have been disclosed. ... Thus, flow charts or source code listings are not a requirement for adequately disclosing the functions of software."); *In re Hayes Microcomputer Prods. Lit.*, 982 F.2d 1527, 1534-35 (Fed. Cir. 1992) (disclosing a microprocessor capable of performing specified functions is sufficient). Here, the Examiner appears to contend that two figures (Figures 4 and 5) showing a microprocessor system capable of performing the recited functions, and two figures providing more detail about the recited functions (Figures 2 and 3, which show the stages), as well as a proffered flow chart consistent with claim 1 as originally filed (Figure 6), are insufficient to satisfy the requirements of 37 CFR 1.83(a) because language not specifically recited in any of the claims is allegedly not illustrated in the drawings. Applicants respectfully disagree and request the Examiner to withdraw this objection to the drawings. If the Examiner continues to maintain the objection to the drawings, a conference with the Examiner is respectfully requested. Each independent claim will be addressed in turn.

Independent claim 1 recites, "[a] method of processing a digital signal by computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) of the digital signal, the method comprising using a multiprocessor computing system having a plurality of processors P configured to perform the steps of: computing an N-point FFT/IFFT of

the signal using first and second sets of butterfly computational stages, each stage in the second set of stages employing a plurality of butterfly operations, wherein each of the butterfly operations in each stage in the second set of stages has a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors." It is unclear what the Examiner means by the contention that the figures do not show the structure of computing. To the extent the Examiner is referring to the multiprocessor system, an embodiment of such a system is shown in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method steps, Figures 2 and 3 illustrate implementations of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22. See also Inventor Decl., ¶ 5. To the extent the Examiner contends "stages" are not shown, Figures 2 and 3 illustrate stages. To the extent the Examiner's contention is that no figure shows "a single, un-nested computation loop," Applicants disagree that any additional figure is needed, however Applicants proffered Figure 6, which the Examiner objected to with a conclusory statement that Figure 6 contained new matter. No new matter was introduced. See original claim 1.

Independent claim 3 recites, "[a] multiprocessor system to process a digital signal by computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) of the signal using a decimation in time or decimation in frequency approach, comprising: the multiprocessor system having a plurality of processors P and configured to implement: means for computing a first plurality of $\log_2 P$ stages of an N-point FFT/IFFT of the signal; means for computing a second plurality of stages of the N-point FFT/IFFT of the signal using in each stage of the second plurality of stages a plurality of butterfly operations, wherein each butterfly operation employs a single butterfly computation loop without employing nested loops; and means for distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stages of the second plurality of stages." It is unclear

what the Examiner means by the contention that the figures do not show the structure of computing. To the extent the Examiner is referring to the multiprocessor system, an embodiment of such a system is shown in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited functions, Figures 2 and 3 illustrate embodiments of the implementation of the functions. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors in the stages. See Page 6, line 25 to 7, line 22. See also Inventor Decl., ¶ 6. To the extent the Examiner's contention is that no figure shows "a single butterfly computational loop without employing nested loops," Applicants disagree that any additional figure is needed, however Applicants proffered Figure 6, which the Examiner objected to with a conclusory statement that Figure 6 contained new matter. No new matter was introduced. See original claim 1.

Independent claim 5 recites, "[a] non-transitory computer-readable storage medium whose contents cause a system having a plurality of processors to perform a linear scalable method of transforming a signal by computing with the plurality of processors a Fast Fourier Transform (FFT) or an Inverse Fast Fourier Transform (IFFT) of the signal, the method comprising: computing a first plurality of stages of an N-point FFT/IFFT; and computing a second plurality of stages of the N-point FFT without employing nested loops and by distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." It is unclear what the Examiner means by the contention that the figures do not show the structure of computing. To the extent the Examiner is referring to a system having a plurality of processors, an embodiment of such a system is shown in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method, Figures 2 and 3 illustrate the implementation of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22. See also Inventor Decl.,

¶ 7. To the extent the Examiner's contention is that no figure shows "computing a second plurality of stages of the N-point FFT without employing nested loops," Applicants disagree that any additional figure is needed, however Applicants proffered Figure 6, which the Examiner objected to with a conclusory statement that Figure 6 contained new matter. No new matter was introduced. See original claim 1.

Independent claim 16 recites, "[a] non-transitory computer-readable storage medium whose contents cause a system having a plurality of processors to perform a linear scalable method of transforming a signal, the method comprising: computing an N-point FFT/IFFT using a first plurality of butterfly computational stages and a second plurality of butterfly computational stages, each stage in the second plurality of stages employing a plurality of butterfly operations having a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." It is unclear what the Examiner means by the contention that the figures do not show the structure of computing. To the extent the Examiner is referring to a system having a plurality of processors, an embodiment of such a system is shown in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method, Figures 2 and 3 illustrate the implementation of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors in the stages. See Page 6, line 25 to 7, line 22. See also Inventor Decl., ¶ 8. To the extent the Examiner's contention is that no figure shows "each stage in the second plurality of stages employing a plurality of butterfly operations having a single, un-nested computation loop," Applicants disagree that any additional figure is needed, however Applicants proffered Figure 6, which the Examiner objected to with a conclusory statement that Figure 6 contained new matter. No new matter was introduced. See original claim 1.

Independent claim 27 recites, "[a] method of transforming a digital signal, the method comprising: using a multiprocessor computing system having a plurality P of processors configured to: compute a first number of butterfly stages of an N-point Fast Fourier Transform

(FFT) or Inverse Fast Fourier transform (IFFT); and compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein each processor computes an equal number of butterfly operations and there is no data dependency between butterflies in a stage of an iteration of the loop." It is unclear what the Examiner means by the contention that the figures do not show the structure of computing. To the extent the Examiner is referring to a multiprocessor computing system having a plurality of processors, an embodiment of such a system is shown in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the recited method, Figures 2 and 3 illustrate the implementation of embodiments of the method. Figure 2 shows a two-processor implementation and Figure 3 shows a four processor implementation, where different line styles represent computation in each of the processors. See Page 6, line 25 to 7, line 22. See also Inventor Decl., ¶ 9. To the extent the Examiner's contention is that no figure shows "compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop," Applicants disagree that any additional figure is needed, however Applicants proffered Figure 6, which the Examiner objected to with a conclusory statement that Figure 6 contained new matter. No new matter was introduced. See original claim 1.

Independent claim 31 recites, "[a] system, comprising: an instruction fetch cache; and a plurality of processors P coupled to the instruction fetch catch and configured to: compute a first number of butterfly stages of an N-point Fast Fourier Transform (FFT) or Inverse Fast Fourier Transform (IFFT) of a digital signal; and compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein there is no data dependency between butterflies in a stage of an iteration of the loop." It is unclear what the Examiner means by the contention that the figures do not show the structure of computing. To the extent the Examiner is referring to a system comprising an instruction fetch cache and a plurality of processors, an embodiment of such a system is shown in Figure 4, where a multiprocessing system comprises clusters 12 (*i.e.*, processors, see Specification at 7, line 24 to 8, line 25). Figure 5 contains more detail on an embodiment of a cluster of Figure 4. To the extent the Examiner is referring to the configuration of the plurality of processors, Figures 2-5 illustrate the implementation of embodiments. Figure 2 shows a two-processor implementation, Figure 3 shows a four processor

implementation, where different line styles represent computation in each of the processors. Figure 4 shows an N processor implementation. See Page 6, line 25 to 7, line 22. See also Inventor Decl., ¶ 10. To the extent the Examiner's contention is that no figure shows "compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop," Applicants disagree that any additional figure is needed, however Applicants proffered Figure 6, which the Examiner objected to with a conclusory statement that Figure 6 contained new matter. No new matter was introduced. See original claim 1.

The Claims Are Enabled by the Specification

The Examiner rejected claims 1-7, 11-13, 15-18, 27-29 and 31-33 under 35 USC Section 112, first paragraph, as not enabled. The Examiner's rejections are respectfully traversed. The Examiner presents claim 1 as an example and contends that "first and second sets of butterfly computational stages ... wherein the second stage employs a single un-nested computational loop," was never fully addressed in the summary or the detail in such a way that one of skill in the art would be able to make or use the invention.

Before addressing the specifics of the claims, it is noted that the Examiner's response does not address the specific arguments and evidence in the form of a declaration presented in the previous amendment of June 21, 2010. For example, the Examiner does not address at all the inventor declaration that was submitted, or the argument that original claim 2 disclosed a specific manner of distributing the remaining stages of the butterfly operations among the processors in an un-nested loop (for example, "by assigning to each processor of the multi-processor system respective addresses of memory locations corresponding to inputs and outputs required for each specific butterfly operation assigned to the processor"). See Inventor Decl., ¶ 12.

Turning to the specifics, there is a strong presumption that the specification contains an adequate disclosure as filed and the Examiner has the burden of presenting evidence or reasons why one of skill in the art would not recognize that the written description provides support for the claims. MPEP 2163(II). Further, there is no *in haec verba* requirement. MPEP 2163. For computer related claims, a disclosure of a processor capable of performing a function and the function is generally sufficient to enable the claims. *See, e.g.*, *Fonar Corp. v. General Electric Corp.*, 107 F.3d 1543, 1549 (Fed. Cir. 1997) ("As a general rule, where software

constitutes part of a best mode of carrying out an invention, description of such a best mode is satisfied by a disclosure of the function of the software. This is because, normally, writing code for such software is within the skill of the art, not requiring undue experimentation, once its functions have been disclosed. ... Thus, flow charts or source code listings are not a requirement for adequately disclosing the functions of software."); *In re Hayes Microcomputer Prods. Lit.*, 982 F.2d 1527, 1534-35 (Fed. Cir. 1992) (One skilled in the art would know how to program a microprocessor to perform the necessary steps described in the specification. Thus, an inventor is not required to describe every detail of his invention. An applicant's disclosure obligation varies according to the art to which the invention pertains. Disclosing a microprocessor capable of performing certain functions is sufficient to satisfy the requirement of section 112, first paragraph, when one of skill in the relevant art would understand what is intended and know how to carry it out.").

Here, the Examiner's complaint appears to be that the specification is concise, and possibly terse. However, that is not the test for enablement. As an initial matter, one of skill in the art would have understood the difference between nested and un-nested loops in general. Inventor Decl., ¶ 12. Thus, one of skill in the art would understand what was meant by an un-nested loop. Figures 2 and 3 show multiple stages of an N-point FFT/IFFT implemented on multiprocessor systems, and the description thereof (see pages 6-8) as well as original claim 2 describe how to implement at least one embodiment of distributing the remaining stages of the butterfly operations among the processors in an un-nested loop (for example, "by assigning to each processor of the multi-processor system respective addresses of memory locations corresponding to inputs and outputs required for each specific butterfly operation assigned to the processor"). See Inventor Decl., ¶ 12. This is sufficient to satisfy the enablement requirement. Thus, claim 1 is enabled. The Examiner has the burden of showing lack of enablement in the context of the language of the claims, and the Examiner has not explained why claim 1 or the other claims are not enabled by the specification. Nevertheless, the remaining claims are enabled for reasons similar to those set forth above with respect to claim 1. Accordingly, all the pending claims are enabled by the specification. Applicants note that they proffered to copy original claims 1 and 2 into the specification, and the Examiner contended (unreasonably) that this would constitute new matter. See original claims 1 and 2.

The Claims Are Sufficiently Definite

The Examiner is thanked for withdrawing the previous rejections of claims 5, 6 and 16-18 under 35 USC Section 112, second paragraph, as indefinite.

The Claims Are Directed to Statury Subject Matter

The Examiner is thanked for withdrawing the previous rejection of claims 1, 2, 7, 11 and 27-29 under 35 USC Section 101, as directed to non-statutory subject matter.

Abel, Alone or in Combination with Jaber, Does Not Render the Claims Obvious

The Examiner rejected claims 1-7, 11-13, 15, 27-29 and 31-33 under 35 USC Section 103(a) as obvious over U.S. Patent No. 5,991,787 issued to Abel et al., in view of U.S. Patent No. 6,792,441 issued to Jaber. The Examiner's rejections are respectfully traversed.

The Examiner appears to rely on Jaber as allegedly teaching the distributing of the butterfly operations. The operation of Jaber for a 16-point FFT distributed among 4 processors is described below with reference to the some of the figures of Jaber, and is contrasted with the operation of the present disclosure. This discussion is illustrative purposes only, and can be generalized to larger FFTs (size N), as well as to different numbers of processors. Inventor Decl., ¶ 14.

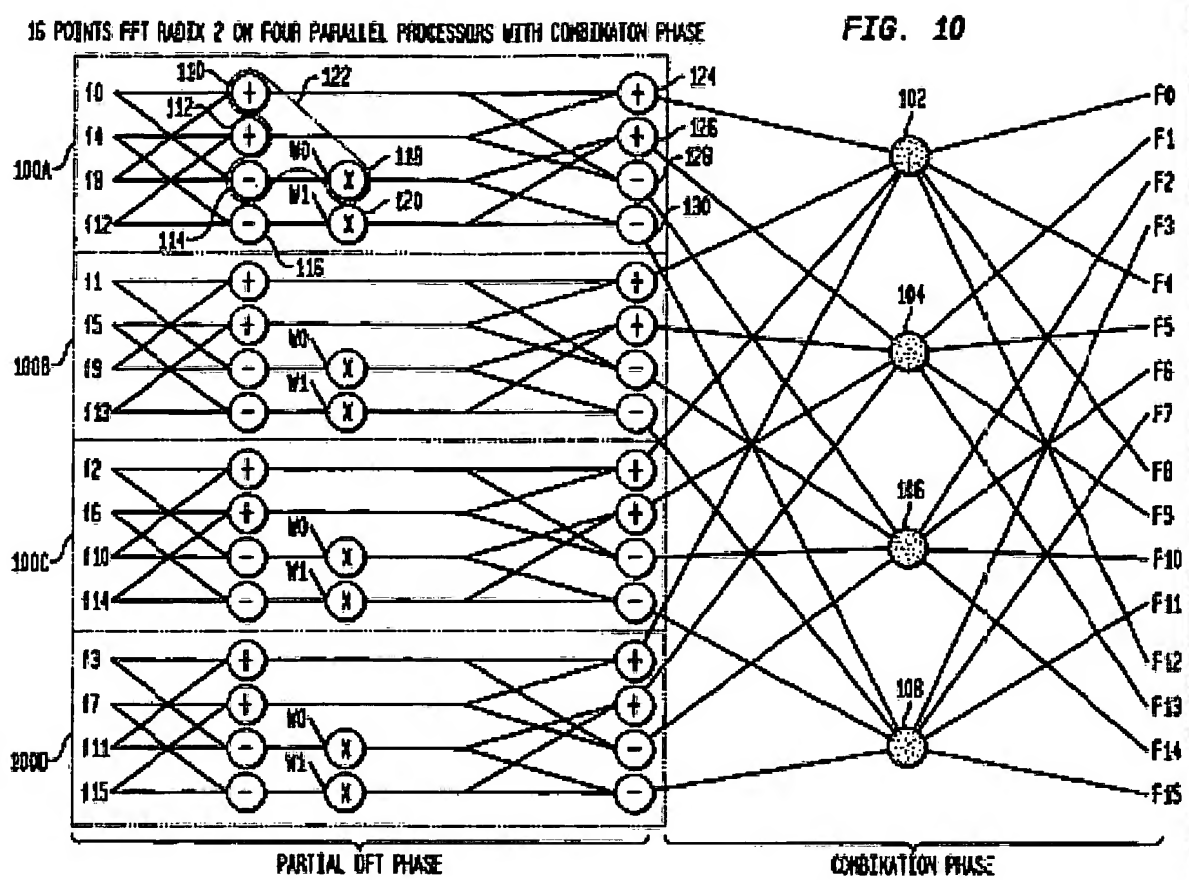


Fig 10 of US6792441 (Jaber et al)

With reference to Figure 10 of Jaber, reproduced above, Jaber teaches that the set of input data points (N=16) is divided among 4 processors (P=4) such that each processor works on 4 data points (i.e. executes 2 butterfly computations) in each stage of an FFT. All 4 processors execute this task in parallel without the need to send or receive data to or from the other parallel processor, for the first two stages of FFT. This is followed by a combination phase which needs the outputs (4x4=16) of all these processors and produces 16 outputs finally. The combination phase is comprised of the last 2 ($\log_2 P$) stages of the FFT. Inventor Decl., ¶ 15.

As a result butterfly computations involving input data points are assigned to the processors as follows:

$$\left.\begin{array}{l}\{f0, f4, f8, f12\} \text{ to first processor} \\ \{f1, f5, f9, f13\} \text{ to second processor} \\ \{f2, f6, f10, f14\} \text{ to third processor} \\ \{f3, f7, f11, f15\} \text{ to fourth processor}\end{array}\right\} \quad G1$$

Inventor Decl., ¶ 16.

Until stage 2 of the FFT computation, all 4 processors execute their butterfly operations in parallel and independently of each other. Thereafter, stages 3 and 4 of the FFT computation are assigned to a combination phase. For the case of a 16-point FFT, the computations require 8 (N/2) twiddle factors/coefficients named, {W0, ..., W7}. The first two stages require the coefficients W0 and W1 only. The combination phase requires the entire set of coefficients {W0,...,W7}. This necessitates that the coefficients W0, W1 be accessible to all the 4 processors at the same time instant. This is evident from Fig 8 of Jaber, reproduced below, which shows a coefficient memory (804) being accessed by all the processors 807A,...807B. Inventor Decl., ¶ 17.

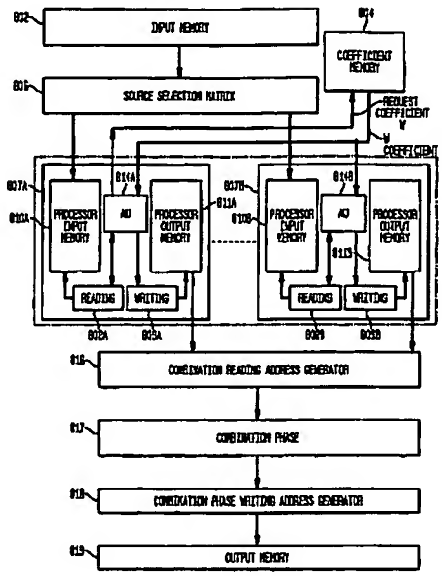


Figure 8 of Jaber

If we consider as a second example the case of a 32 point FFT (N=32) on 4 parallel processors (P=4), input data points are assigned to the processors as follows:

$$\left.\begin{array}{l}\{f0, f4, f8, f12, f16, f20, f24, f28\} \text{ to first processor} \\ \{f1, f5, f9, f13, f17, f21, f25, f29\} \text{ to second processor} \\ \{f2, f6, f10, f14, f18, f22, f26, f30\} \text{ to third processor} \\ \{f3, f7, f11, f15, f19, f23, f27, f31\} \text{ to fourth processor}\end{array}\right\} \quad G2$$

Inventor Decl., ¶ 18.

There are total 5 stages in this case. Until stage 3 of the FFT computation, all 4 processors execute their butterfly operations in parallel and independently of each other. Thereafter, stages 4 and 5 of the FFT computation are assigned to a combination phase. In this case too, the combination phase is comprised of the last $\log_2 P = 2$ stages of the 32 point FFT. Inventor Decl., ¶ 19.

In this case there are 16 twiddle coefficients {W0,…,W15}, out of which {W0,W1,W2,W3} are required to be accessible to all the 4 processors. The combination phase requires the entire set of coefficients {W0,…,W15}. Inventor Decl., ¶ 20.

In the general case of an N point FFT being executed on P parallel processors, there would be a total of $\log_2 N$ stages, out of which the last $\log_2 P$ stages comprise the combination phase and the remaining $(\log_2 N - \log_2 P)$ stages are computed in P processors in parallel. In the general case, there would be N/2 twiddle coefficients in all $\{W0,\ldots,W_{N/2}-1\}$, out of which the identical N/(2P) number of twiddle coefficients need to be accessible to all the P parallel processors all the while. The combination phase requires access to all the N/2 twiddle coefficients. Inventor Decl., ¶ 21.

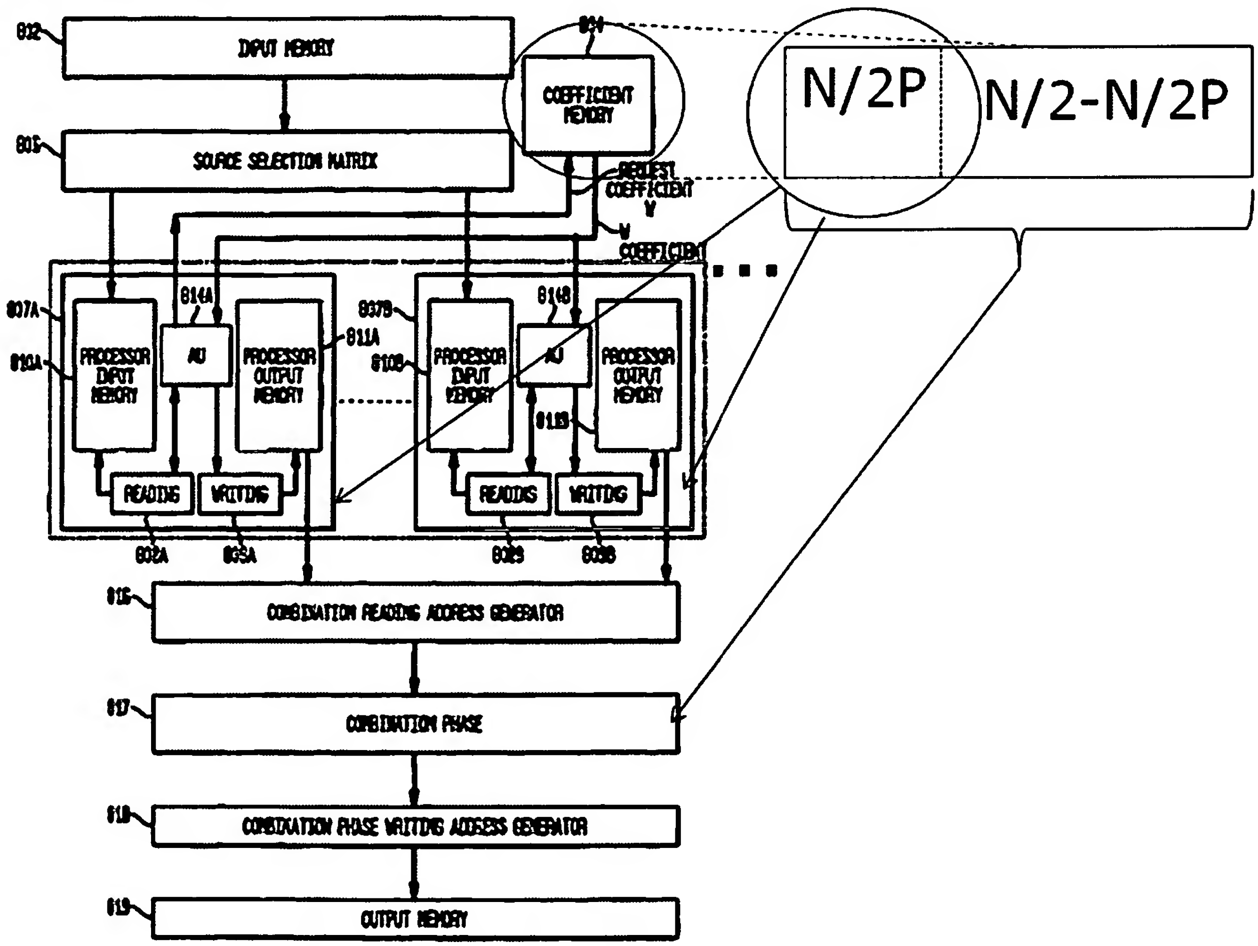
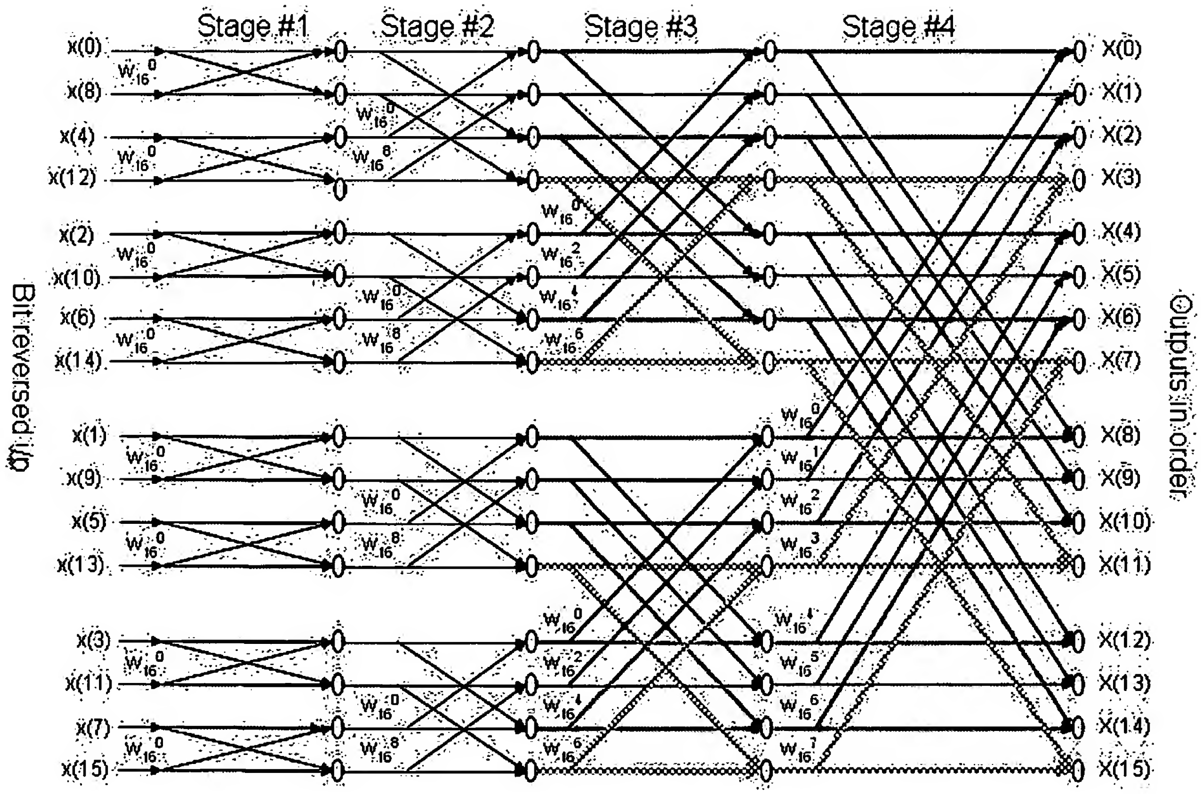


Figure 8 of Jaber Modified to Show Distribution of Twiddle Coefficients in Jaber

Figure 8 of Jaber appears again above, modified to show the distribution of the twiddle coefficients from the coefficient memory as per the Jaber. Out of N/2 coefficients, N/2P coefficients are needed by all of the processors all of the time, and the entire set is needed by the combinational phase. Inventor Decl., ¶ 22.



Butterfly Distribution of Present Disclosure for 4-Processor Configuration: N = 16; P = 4

The butterfly distribution of the present disclosure is illustrated above for N = 16 and P = 4. The set of input data points (N=16) is divided among 4 processors (P=4) such that each processor works on 4 data points (i.e. executes 2 butterfly computations) in each stage of the FFT. (A 16 point FFT is shown to facilitate a comparison with Jaber; Figure 3 of the present application shows an 8 point FFT on a 4 processor system). All 4 processors execute this task in parallel without the need to send or receive data to or from the other parallel processors, for the last two stages of the FFT. During the first $\log_2 P$ =2, there is data dependency between the parallel processors, *i.e.* there is need of data to/from each other in the first two stages. Therefore, parallel processing of butterfly computations starts after the first 2 stages. As shown in the above figure (stage 3, stage 4), the butterflies colored in red are computed in the first processor, the butterflies colored in blue are computed in the second processor, the butterflies colored in green are computed in the third processor, the butterflies colored in black are computed in the fourth processor. It can be seen from stage 4 that all the red colored butterflies take input data points only from red colored butterflies of the previous stage (i.e. stage 3), and so forth for the other

processors. Inventor Decl., ¶ 23. As a result butterfly computations involving input data points are assigned to the processors as follows:

$$\left.\begin{array}{l}\{x(0),x(8),x(1),x(9)\} \text{ to first processor}\\ \{x(4), x(12), x(5), x(13)\} \text{ to second processor}\\ \{x(2), x(10), x(3), x(11)\} \text{ to third processor}\\ \{x(6), x(14), x(7), x(15)\} \text{ to fourth processor}\end{array}\right\} \quad G3$$

Inventor Decl., ¶ 23.

After stage 2 of the FFT computation, all 4 processor execute their butterfly operations in parallel and independently of each other until the final output. Prior to that, stages 1 and 2 of the FFT computation have data dependency among the parallel processors. For the case of a 16-point FFT, the computations require 8 (=N/2) twiddle factors/coefficients $\{W^0,\ldots,W^7\}$. Inventor Decl., ¶ 24. As shown in the butterfly distribution illustrated on the previous page, the requirement of twiddle coefficients by the 4 processors is as follows:

$$\left.\begin{array}{l}\{w^0, w^4\} \text{ to first processor}\\ \{w^1, w^5\} \text{ to second processor}\\ \{w^2, w^6\} \text{ to third processor}\\ \{w^3, w^7\} \text{ to fourth processor}\end{array}\right\} \quad G4$$

Inventor Decl., ¶ 24.

It may be noted that owing to the innovative scheme of distribution of butterfly computations among processors, the sets of twiddle coefficients required by different processors are disjoint and the number of coefficients in each set does not exceed 2. Inventor Decl., ¶ 25.

If we consider as a second example the case of a 32 point FFT (N=32) on 4 parallel processors (P=4), input data points are assigned to the processors as follows:

$$\left.\begin{array}{l}
\{x(0),x(16),x(1),x(17), x(2),x(18),x(3),x(19)\} \text{ to first processor} \\
\{x(8),x(24),x(9),x(25), x(10),x(26),x(11),x(27)\} \text{ to second processor} \\
\{x(4),x(20),x(5),x(21), x(6),x(22),x(7),x(23)\} \text{ to third processor} \\
\{x(12),x(28),x(13),x(29), x(14),x(30),x(15),x(31)\} \text{ to fourth processor}
\end{array}\right\} \text{G5}$$

Inventor Decl., ¶ 26.

After stage 2 of the FFT computation, all 4 processor execute their butterfly operations in parallel and independently of each other until the final output. Prior to that, stages 1 and 2 of the FFT computation have data dependency among the parallel processors. For the case of a 32-point FFT, the computations require 16 (=N/2) twiddle factors/coefficients named $\{W^0,\ldots,W^{15}\}$. The requirement of twiddle coefficients by the 4 processors is as follows:

$$\left.\begin{array}{l}
\{w^0, w^4, w^8, w^{12}\} \text{ to first processor} \\
\{w^1, w^5, w^9, w^{13}\} \text{ to second processor} \\
\{w^2, w^6, w^{10}, w^{14}\} \text{ to third processor} \\
\{w^3, w^7, w^{11}, w^{15}\} \text{ to fourth processor}
\end{array}\right\} \text{G6}$$

Inventor Decl., ¶ 27.

It may be noted again that owing to the innovative scheme of distribution butterfly computations among processors, the sets of twiddle coefficients required by different processors are disjoint and the number of coefficients in each set does not exceed 4. Inventor Decl., ¶ 28.

In the general case of an N point FFT being executed on P parallel processors, there would be a total of $\log_2 N$ stages, out of which the first $\log_2 P$ stages have data dependency between the parallel processors and the remaining ($\log_2 N$- $\log_2 P$) stages are computed in P processors in parallel without need of data to/from each other. In the general case, there would

be N/2 twiddle coefficients in all $\{W^0,\ldots,W^{N/2-1}\}$, out of which disjoint sets N/(2P) of twiddle coefficients are needed by each of the P parallel processors except when P=2. In this case also (P=2), one processor only needs all the twiddle coefficients, whereas the other one needs only a subset having N/(2P) twiddle coefficients. Inventor Decl., ¶ 29.

The fundamental difference lies in the method of butterfly distribution among the parallel processor as shown in the assignments of butterflies among the parallel processors, G1, G3 for 16-point FFT/IFFT and G2, G5 for 32-point FFT/IFFT. As a result of this, the stages where there are dependencies among the parallel processors are in the first $\log_2 P$ stages as per the present disclosure, whereas, the dependencies among the parallel processors are in the last $\log_2 P$ stages as per Jaber. Inventor Decl., ¶ 30.

Another manifestation of the present disclosure can be seen in the uses of twiddle coefficients in the different parallel processors. As per the present proposal, it may be noted that although all N/2 coefficients are used for the entire FFT computation, no parallel processor uses more than an N/2P subset of these, having no common coefficients with any other processors except when P=2. Whereas as per Jaber, an identical N/(2P) twiddle coefficients need to be accessible to all the P parallel processors all the while. The combination phase of Jaber requires access to all the N/2 twiddle coefficients. Figure 8 of Jaber is modified again on the following page to show how the twiddle coefficients would be applied if Jaber were modified in accordance with the disclosure of the present application. As can be seen, distinct N/2P subsets of the N/2 coefficients in the coefficient memory are used by different parallel processors. At any stage no processor has need of any coefficients which are required by any other processor. Please note that the present disclose is not limited in any way to application to the particular architecture of Jaber. It can be applicable to any system architecture in general, including shared and distributed memory systems. The use of shared coefficient memory taught in Jaber is rendered unnecessary in the present proposal due to the innovative method of distribution of butterflies among the parallel processors. Inventor Decl., ¶ 31.

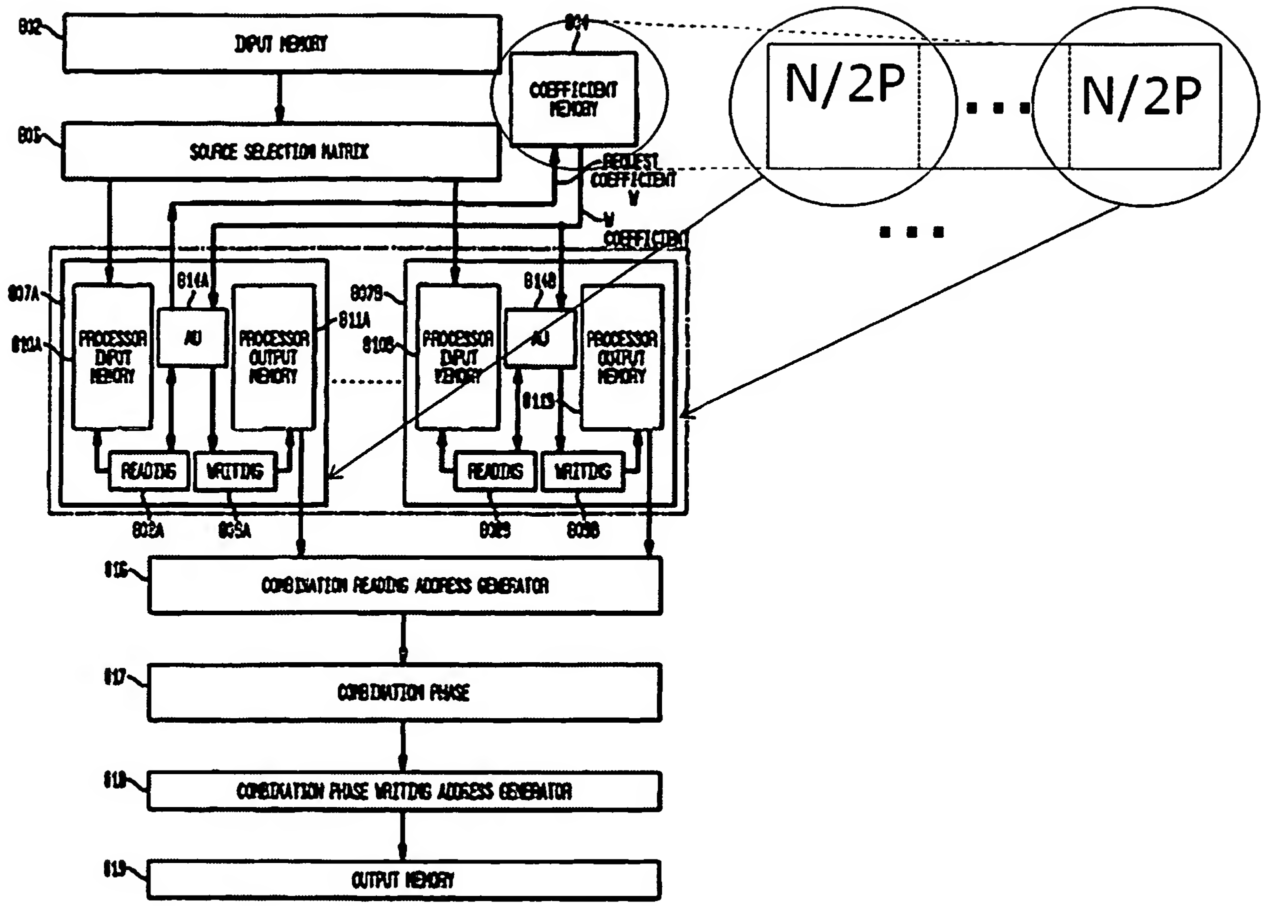


Figure 8 of Jaber Modified to Show Distribution of Twiddle Coefficients
According to Present Disclosure

Turning to the language of the claims, claim 1 recites, “each stage in the second set of stages has a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors.” Abel, alone or in combination with Jaber, does not teach or suggest, or otherwise render obvious, “distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors,” as recited. Inventor Decl., ¶ 32. Claims 2, 7 and 11 depend from claim 1 and are allowable at least by virtue of their dependencies.

Previously, Applicants compared and contrasted the operation of Jaber and the present disclosure as background for its argument that Jaber did not teach a recited limitation as contended by the Examiner. For example, with respect to claim 1, that Jaber did not teach "distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors," as contended by the Examiner. The Examiner also contends in response that because Jaber stores the coefficients in memory and retrieves the coefficients from memory there is no data dependency. This argument is incorrect because in Jaber, the coefficients are shared between the processors, as discussed above. Specifically, out of N/2 coefficients, N/2P coefficients are needed by all of the processors of Jaber all of the time, and the entire set is needed by the combinational phase. Thus, the stages of Jaber have data interdependencies among the processors. Inventor Decl., ¶ 33.

In response, the Examiner points to Figures 5A and 5B of Jaber, which the Examiner contends shows the modules of Jaber do not require data/coefficients from other parallel modules. It is not seen how Figures 5A and 5B of Jaber establish an alleged lack of data interdependency in Jaber, let alone the recited "distributing the plurality of butterfly operations in each stage of the second set of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency among the parallel processors."

Independent claim 3 recites, "means for computing a second plurality of stages of the N-point FFT/IFFT of the signal using in each stage of the second plurality of stages a plurality of butterfly operations, wherein each butterfly operation employs a single butterfly computation loop without employing nested loops; and means for distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stages of the second plurality of stages." Abel, alone or in combination with Jaber, does not teach or suggest, or otherwise render obvious, "means for distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stages of the second plurality of stages," as recited. Inventor Decl., ¶ 34. Claims 4, 12, 13 and 15 depend

from claim 3 and are allowable at least by virtue of their dependencies. With regard to the Examiner's responses, it is noted that it was argued that a specific recited feature is missing from the cited combination of references, and, as discussed above with respect to claim 1, the stages of Jaber have data interdependencies among the processors.

Independent claim 5, as amended, recites, "computing a first plurality of stages of an N-point FFT/IFFT; and computing a second plurality of stages of the N-point FFT without employing nested loops and by distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." Abel, alone or in combination with Jaber, does not teach or suggest, or otherwise render obvious, "computing a second plurality of stages of the N-point FFT without employing nested loops and by distributing the butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage," as recited. Inventor Decl., ¶ 35. Claim 6 depends from claim 5 and is allowable at least by virtue of its dependencies. With regard to the Examiner's responses, it is noted that it was argued that a specific recited feature is missing from the cited combination of references, and, as discussed above with respect to claim 1, the stages of Jaber have data interdependencies among the processors.

Independent claim 16, as amended, recites, "computing an N-point FFT/IFFT using a first plurality of butterfly computational stages and a second plurality of butterfly computational stages, each stage in the second plurality of stages employing a plurality of butterfly operations having a single, un-nested computation loop; and distributing the plurality of butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage." Abel, alone or in combination with Jaber, does not teach, suggest or otherwise render obvious, "distributing the plurality of butterfly operations in each stage of the second plurality of stages such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage," as recited. Inventor Decl., ¶ 36. Claims 17 and 18 are allowable at least by virtue of their dependencies. With regard to the Examiner's responses, it is noted that it was argued that a specific recited feature is

missing from the cited combination of references, and, as discussed above with respect to claim 1, the stages of Jaber have data interdependencies among the processors.

Independent claim 27, as amended, recites, "compute a first number of butterfly stages of an N-point Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT); and compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein each processor computes an equal number of butterfly operations and there is no data dependency between butterflies in a stage of an iteration of the loop." Abel, alone or in combination with Jaber, does not teach, suggest or otherwise render obvious, "compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein each processor computes an equal number of butterfly operations and there is no data dependency between butterflies in a stage of an iteration of the loop," as recited. Inventor Decl., ¶ 37. Claims 28 and 29 are allowable at least be virtue of their dependencies. With regard to the Examiner's responses, it is noted that it was argued that a specific recited feature is missing from the cited combination of references, and, as discussed above with respect to claim 1, the stages of Jaber have data interdependencies among the processors.

Independent claim 31, as amended, recites, "compute a first number of butterfly stages of an N-point Fast Fourier Transform (FFT) or Inverse Fast Fourier Transform (IFFT) of a digital signal; and compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein there is no data dependency between butterflies in a stage of an iteration of the loop." Abel, alone or in combination with Jaber, does not teach, suggest or otherwise render obvious, "compute remaining butterfly stages of the N-point FFT/IFFT with a single iterative loop wherein there is no data dependency between butterflies in a stage of an iteration of the loop," as recited. Inventor Decl., ¶ 38. Claims 32 and 33 are allowable at least by virtue of their dependencies. With regard to the Examiner's responses, it is noted that it was argued that a specific recited feature is missing from the cited combination of references, and, as discussed above with respect to claim 1, the stages of Jaber have data interdependencies among the processors.

<u>No New Matter Has Been Introduced</u>

The Examiner also has consistently objected to the claims in a conclusory manner as allegedly introducing new matter. Specifically, the Examiner previously contended each of

the independent claims introduces new matter based on the limitations "first and second sets of butterfly computational stages ... wherein the second stage employs single un-nested computational loop." Applicants pointed out that none of the independent claims contained this language. In response, the Examiner now contends that the limitation "wherein each of the butterfly operations in each stage in the second set of stages has a single, un-nested computational loop," which appears in independent claim 1, but which does not appear in the other independent claims, allegedly introduces new matter. The Examiner does explain why this language allegedly introduces new matter and does not address the specific arguments previously presented with respect to each of the independent claims that no new matter had been introduced. The Examiner's objections are respectfully traversed.

As an initial matter, only one of the current independent claims contains the precise language objected to by the Examiner. Thus, it is difficult, if not impossible, for Applicants to ascertain what exactly the Examiner considers to be new matter in the other independent claims, and the Examiner does not explain the basis for the objection with respect to any of the claims. Nevertheless, Applicants will attempt to address each independent claim in turn.

Independent claim 1 as originally filed appears below.

1. A linear scalable method for computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) in a multiprocessing system using a decimation in time approach, comprising the steps of:

computing first and second stages of $\log_2 N$ stages of an N-point FFT/IFFT as a single radix-4 butterfly operation while implementing the remaining ($\log_2 N$-2) stages using radix-2 butterfly operations, wherein each radix-2 butterfly operation employs a single radix-2 butterfly computation loop without employing nested loops; and

distributing the butterfly operations in each stage such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage.

There is a strong presumption that the specification contains an adequate disclosure as filed and the Examiner has the burden of presenting evidence or reasons why one of skill in the art would not recognize that the written description provides support for the claims. MPEP 2163(II). Further, there is no *in haec verba* requirement. MPEP 2163. *Id.* If it is the word "sets" to which the Examiner objects, claim 1 as originally filed clearly was addressed to processing two sets of stages (the first set comprising the first and second stages of butterfly computational stages and the second set comprising the remaining stages), with the second set (the remaining stages) employing "a single ... computational loop without employing nested loops." A "set" has a well-known meaning and one of skill in the art would have recognized that original claim 1 referred to sets of computational stages. Inventor Decl., ¶ 42. Moreover, Figures 2 and 3 illustrates sets of computational stages and the description thereof on pages 6-7 is clearly directed to handling the first two stages in one manner and the remaining stages in another. *Id.* To the extent the Examiner is referring to "a single un-nested computational loop," this is almost identical to the original language "a single ... computational loop without employing nested loops." Further, as discussed above, one of skill in the art would have recognized Figures 2 and 3 and the description thereof on pages 6 and 7 as disclosing "computing an N-point FFT/IFFT of the signal using first and second sets of butterfly computational stages, each set in the second set of stages employing a plurality of butterfly operations, wherein each of the butterfly operations in each stage in the second set of stages has a single, un-nested computational loop." Inventor Decl., ¶ 43. Thus, it is not seen how claim 1 introduces any new matter. To the extent the Examiner is really asserting an enablement rejection, this rejection is addressed above.

Independent claim 3, as originally filed, appears below.

3. A linear scalable system for computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) in a multiprocessing system using a decimation in time approach, comprising:

means for computing first and second stages of $\log_2 N$ stages of an N-point FFT/IFFT as a single radix-4 butterfly operation while implementing the remaining ($\log_2 N$-2)

stages using radix-2 butterfly operations, wherein each radix-2 butterfly operation employs a single radix-2 butterfly computation loop without employing nested loops; and

means for distributing the butterfly operations in each stage such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage.

As noted above, there is no *in hac verba* requirement. Claim 3 as previously amended does not use the word "sets." If it is "plurality of stages" to which the Examiner objects, it is respectfully submitted that "plurality" has a definite meaning in the claims of a patent, specifically, more than one, and one of skill in the art would have recognized both "first and second stages" as a first plurality of stages, and "remaining stages" as a second plurality of stages. Inventor Decl., ¶ 46. If it is "employs a single ... butterfly computational loop without employing nested loops," to which the Examiner objects, this language appears in claim 3 as originally filed. Thus, it is not seen how claim 3 as previously amended introduces any new matter. To the extent the Examiner is really asserting an enablement rejection, this rejection is addressed above.

Independent claim 5, as originally filed, appears below.

5. A computer program product comprising computer readable program code stored on a computer readable storage medium embodied therein for computing a Fast Fourier Transform (FFT) or Inverse Fast Fourier transform (IFFT) in a multiprocessing system using a decimation in time approach, comprising:

computer readable program code means configured for computing computing first and second stages of $\log_2 N$ stages of an N-point FFT/IFFT as a single radix-4 butterfly operation while implementing the remaining ($\log_2 N$-2) stages using radix-2 butterfly operations, wherein each radix-2 butterfly operation employs a single radix-2 butterfly computation loop without employing nested loops; and

computer readable program code means configured for distributing the butterfly operations in each stage such that each processor computes an equal number of complete butterfly operations thereby eliminating data interdependency in the stage.

As noted above, there is no *in hac verba* requirement. Claim 5 as previously amended does not use the word "sets." If it is "plurality of stages" to which the Examiner objects, it is respectfully submitted that "plurality" has a definite meaning in the claims of a patent, specifically, more than one, and one of skill in the art would have recognized both "first and second stages" as a first plurality of stages, and "remaining stages" as a second plurality of stages. Inventor Decl., ¶ 50. If it is "without employing nested loops," to which the Examiner objects, this language appears in claim 5 as originally filed. Thus, it is not seen how claim 5 as previously amended introduces any new matter. To the extent the Examiner is really asserting an enablement rejection, this rejection is addressed above.

With regard to independent claim 16, to the extent the Examiner objects to the word "plurality," it is respectfully submitted that "plurality" has a definite meaning in the claims of a patent, specifically, more than one, and one of skill in the art would have recognized both "first and second stages" (see original claim 5) as a first plurality of stages, and "remaining stages" (see original claim 5) as a second plurality of stages. Inventor Decl., ¶ 52. To the extent the Examiner objects to "having a single, un-nested computational loop," Applicants refer the Examiner to original claims 1, 3 and 5, and the discussion of claims 1, 3 and 5 above. Thus, it is not seen how claim 16 as previously amended introduces any new matter. To the extent the Examiner is really asserting an enablement rejection, this rejection is addressed above.

With regard to independent claims 27 and 31, to the extent the Examiner objects to "a first number of butterfly stages" it is respectfully submitted that one of skill in the art would have recognized "first and second stages" (see original claim 5) as a first number of stages, and "remaining stages" (see original claim 5) as "remaining stages." Inventor Decl., ¶ 53. To the extent the Examiner objects to "a single iterative loop," Applicants refer the Examiner to the specification, page 7, lines 8-13. Thus, it is not seen how claims 27 and 31 as previously amended introduce any new matter. To the extent the Examiner is really asserting enablement rejections, these rejections are addressed above.

The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC

Timothy L. Boller
Registration No. 47,435

TLB:bmw

701 Fifth Avenue, Suite 5400
Seattle, Washington 98104
Phone: (206) 622-4900
Fax: (206) 682-6031

1736443_1.DOC